AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all prior versions and listings of claims:

Listing of Claims:

1. (Previously presented) An encoder for use in a nonvolatile counter comprising:

a logic gate having a first input for receiving an up/down signal, a second input for receiving a most-significant bit signal, and an output;

a plus-one block having an input for receiving an inverted input signal and an output; and

a two-to-one multiplexer having a first input for receiving the inverted input signal, a second input coupled to the output of the plus-one block, a control input coupled to the output of the logic gate, and an output for providing an encoded signal.

2. (Original) The encoder of claim 1 in which the first input is set to a logic one for counting up.

3. (Original) The encoder of claim 1 in which the first input is set to a logic zero for counting down.

4. (Original) The encoder of claim 1 in which the logic gate comprises a XNOR gate.

5. (Original) The encoder of claim 1 in which the plus-one block comprises a combinatorial logic circuit.

6. (Original) The encoder of claim 1 in which the encoded signal is equal to the inverted input signal if the control signal is a logic zero.

7. (Original) The encoder of claim 1 in which the encoded signal is equal to the inverted input signal plus one if the control signal is a logic one.

8. (Original) The encoder of claim 1 in which the nonvolatile counter comprises a ferroelectric nonvolatile counter.

9. (Original) The encoder of claim 1 in which the nonvolatile counter comprises an *m*-bit nonvolatile counter.

10. (Original) The encoder of claim 9 in which *m* is equal to 4.

11. (Original) The encoder of claim 9 in which *m* is equal to 5.

12. (Original) The encoder of claim 9 in which *m* is equal to 40.

13-14. (Cancelled)

15. (Previously presented) A data encoding method for counting up from input number data $n = d_m d_{m-1} d_{m-2} \ldots d_1 d_0$ comprising:

adding a logic one to inverted input number data to provide intermediate number data;

multiplexing the inverted input number data and the intermediate number data to provide output number data; such that:

if *n* is even, then the output number data comprises $n+1 = \overline{d}_m \overline{d}_{m-1} \overline{d}_{m-2} \ldots \overline{d}_1 \overline{d}_0$; and

if *n* is odd, then the output number data comprises $n+1 = \overline{d}_m \overline{d}_{m-1} \overline{d}_{m-2} \ldots \overline{d}_1 \overline{d}_0 + 1$.

16. (Previously presented) The data encoding method of claim 15 further comprising keeping an output count at a maximal number when counting up from the maximal number instead of rolling the output count over to zero.

17. (Previously presented) The data encoding method of claim 16 in which the maximal number is equal to 1000 when *m* is equal to four.

18. (Previously presented) A data encoding method for counting down from [[an]] input number data $n = d_m d_{m-1} d_{m-2} \ldots d_1 d_0$ comprising:

adding a logic one to inverted input number data to provide intermediate number data;

multiplexing the inverted input number data and the intermediate number data to provide output number data; such that:

if n is even, then the output number data comprises $n-1 = \bar{d}_m \bar{d}_{m-1} \bar{d}_{m-2} \ldots \bar{d}_1 \bar{d}_0 + 1$; and

if n is odd, then the output number data comprises $n-1 = \bar{d}_m \bar{d}_{m-1} \bar{d}_{m-2} \ldots \bar{d}_1 \bar{d}_0$.

19. (Previously presented) The data encoding method of claim 18 further comprising keeping an output count at zero when counting down from zero, instead of rolling the output count over to a maximal number.

20. (Previously presented) The data encoding method of claim 18 in which the maximal number is equal to 1000 when m is equal to four.